



**UNIVERSIDAD  
DE GRANADA**

**Ingeniería de  
Sistemas de Información**

**Casos Prácticos**

**Ingeniería Informática, curso 2020-2021  
Departamento de Ciencias de la Computación  
e Inteligencia Artificial**

# Casos prácticos

Este documento describe el trabajo que ha de realizarse para los seminarios de la asignatura de Ingeniería de Sistemas de Información, en los que se estudiarán diferentes casos prácticos sobre el uso de sistemas de información en diferentes dominios de aplicación.

El objetivo del trabajo es analizar con detalle algún aspecto relevante de las arquitecturas software que se han diseñado para resolver los problemas específicos asociados a distintos tipos de sistemas de información. Para ello, cada alumno deberá seleccionar un tema de la lista propuesta (o proponerle algún tema al profesor de la asignatura), leer como mínimo las referencias que se proporcionan para el tema que le haya sido asignado y buscar información sobre los sistemas actuales que se utilizan en el ámbito analizado. El resultado del trabajo deberá entregarse en formato electrónico mediante una memoria y exponerse en público durante las sesiones de clase previstas para ello.

## Normas de realización del trabajo

En primer lugar, se debe escoger uno de los temas propuestos, que ya han sido analizados con cierto detalle en publicaciones científicas (aquéllas que requieren un proceso de revisión por pares, o *peer review*, para la publicación de un trabajo, tales como revistas de investigación con “índice de impacto” o congresos organizados por asociaciones de reconocido prestigio como la ACM o IEEE Computer Society).

Una vez elegido el tema del trabajo, se deberán analizar con detalle las publicaciones indicadas para cada tema y utilizar dichas publicaciones como punto de partida para comprender qué enfoques y técnicas se utilizan para resolver los problemas que se plantean cuando uno pretende diseñar un sistema de información adecuado para satisfacer las necesidades particulares de distintos tipos de aplicaciones.

En particular, se deberá prestar atención a las arquitecturas que se proponen, sus características no funcionales (rendimiento, escalabilidad...) y las decisiones de diseño por las que se ha optado al construir sistemas de información reales para resolver problemas de interés práctico. Como mínimo, se deberán analizar tres publicaciones, a ser posible de distintos autores, en las que se aborde el mismo tipo de problema desde ángulos diferentes o se introduzcan mejoras significativas con respecto a las soluciones propuestas con anterioridad.

NOTA: La restricción del número mínimo de trabajos se deriva de la regla del 3 de Jerry Weinberg (*The Secrets of Consulting: A Guide to Giving and Getting Advice Successfully*), que se puede plantear de distintas maneras:

- Before deciding on a course of action, come up with three alternatives.
- If you can't think of three things that might go wrong with your plans, then there's something wrong with your thinking.
- If you don't have three options for a solution to a problem, you don't understand it well enough yet, and you might need to explore it more.

## Documentación y entrega del trabajo

- Cada alumno abordará un tema diferente, por lo que, antes de ponerse a trabajar en él, deberá consultarlo con el profesor de la asignatura, que irá asignando los temas por orden de llegada. Si el tema propuesto ya ha sido asignado previamente, deberá buscar un tema alternativo sobre el que desarrollar su trabajo.
- Deberá elaborar un documento, que puede ser el mismo que luego utilice en su presentación, en el que se describa el análisis realizado y se recojan correctamente todas las referencias bibliográficas empleadas en su estudio. Este documento puede ir acompañado de cuantos apéndices considere oportunos para describir aquellos detalles del trabajo que no aparezcan debidamente reflejados en su presentación. El documento o presentación, junto a sus posibles apéndices, deberá entregarse en formato PDF antes de su exposición en clase.
- El trabajo realizado para esta parte del curso se expondrá en las sesiones de clase previstas en el calendario del curso. La presentación de cada trabajo se realizará mediante una exposición, de una duración no superior a diez minutos, seguida de un turno de preguntas, en el que el autor de cada trabajo responderá a las cuestiones planteadas por sus compañeros y atenderá a las sugerencias que éstos puedan hacerle sobre formas alternativas de diseñar sistemas de información para abordar los problemas estudiados.

## Evaluación del trabajo

- 50 % por la documentación aportada al entregar el trabajo.
- 50 % por la presentación realizada durante las sesiones de laboratorio y la participación en las discusiones que se desarrollen.

---

## Temas propuestos

### Fabricación de circuitos integrados

- Rogier Wester, John Koster: The Software behind Moore's Law. IEEE Software 32(2):37-40, March/April 2015 DOI <http://dx.doi.org/10.1109/MS.2015.53>

### Sistemas distribuidos: MapReduce

- Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: Simplified data processing on large clusters. Communications of the ACM 51, 1 (January 2008), 107-113. DOI <http://dx.doi.org/10.1145/1327452.1327492>

### Sistemas distribuidos: Spark

- Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. 2016. Apache Spark: A unified engine for big data processing. Communications of the ACM 59, 11 (October 2016), 56-65. DOI: <https://doi.org/10.1145/2934664>

### Sistemas distribuidos: Big Data

- Ian Gorton, John Klein: Distribution, Data, Deployment: Software Architecture Convergence in Big Data Systems. IEEE Software 32(3):78-85, May/June 2015 DOI <http://dx.doi.org/10.1109/MS.2014.51>

### Sistemas basados en flujos de datos: TensorFlow (Google)

- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, Xiaoqiang Zheng: TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arXiv e-prints, arXiv:1603.04467, March 2016. <https://arxiv.org/abs/1603.04467>
- Peter Goldsborough: A Tour of TensorFlow, arXiv e-prints, arXiv:1610.01178, October 2016. <https://arxiv.org/abs/1610.01178>

---

## Sistemas basados en flujos de datos: Naiad (Microsoft)

- Derek G Murray, Frank McSherry, Rebecca Isaacs, Michael Isard, Paul Barham, Martín Abadi: Naiad: a timely dataflow system. In Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles, SOSP'2013, pages 439–455. ACM, 2013. <http://sigops.org/s/conferences/sosp/2013/papers/p439-murray.pdf>
- Derek G. Murray, Frank McSherry, Michael Isard, Rebecca Isaacs, Paul Barham, Martin Abadi. 2016. Incremental, iterative data processing with timely dataflow. Communications of the ACM 59, 10 (September 2016), 75-83. DOI: <https://doi.org/10.1145/2983551>
- Johannes Gehrke. 2016. Technical Perspective: Naiad. Communications of the ACM 59, 10 (September 2016), 74-74. DOI: <https://doi.org/10.1145/2985784>

## Sistemas basados en flujos de datos: Halide

- Jonathan Ragan-Kelley, Connelly Barnes, Andrew Adams, Sylvain Paris, Frédo Durand, and Saman Amarasinghe. 2013. Halide: a language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines. In Proceedings of the 34th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '13). ACM, New York, NY, USA, 519-530. DOI: <https://doi.org/10.1145/2491956.2462176>
- Jonathan Ragan-Kelley, Andrew Adams, Dillon Sharlet, Connelly Barnes, Sylvain Paris, Marc Levoy, Saman Amarasinghe, and Frédo Durand. 2017. Halide: decoupling algorithms from schedules for high-performance image processing. Communications of the ACM 61, 1 (January 2018), 106-115. DOI: <https://doi.org/10.1145/3150211>

## Sistemas basados en flujos de datos: Apache Kafka

- Ken Goodhope, Joel Koshy, Jay Kreps, Neha Narkhede, Richard Park, Jun Rao, Victor Yang Ye: Building LinkedIn's Real-time Activity Data Pipeline. IEEE Data Engineering Bulletin, 35(2):33–45, 2012. <http://sites.computer.org/debull/A12june/pipeline.pdf>
- Guozhang Wang, Joel Koshy, Sriram Subramanian, Kartik Paramasivam, Mammad Zadeh, Neha Narkhede, Jun Rao, Jay Kreps, and Joe Stein. 2015. Building a replicated logging system with Apache Kafka. Proceedings of the VLDB Endowment - Proceedings of the 41st International Conference on Very Large Data Bases, 8, 12 (August 2015), 1654-1655. DOI <http://dx.doi.org/10.14778/2824032.2824063>

---

## Búsqueda en Internet

- Mike Andrews. 2012. Searching the Internet. IEEE Software 29, 2 (March 2012), 13-16. DOI <http://dx.doi.org/10.1109/MS.2012.39>
- Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems 30, 1-7 (April 1998), 107-117. DOI [http://dx.doi.org/10.1016/S0169-7552\(98\)00110-X](http://dx.doi.org/10.1016/S0169-7552(98)00110-X)

## Software multimedia para dispositivos móviles

- Lionel Bouchard. 2010. Multimedia Software for Mobile Phones. IEEE Software 27, 3 (May 2010), 8-10. DOI <http://dx.doi.org/10.1109/MS.2010.78>

## Dispositivos médicos

- Lennart Hofland and Joop van der Linden. 2010. Software in MRI Scanners. IEEE Software 27, 4 (July 2010), 87-89. DOI <http://dx.doi.org/10.1109/MS.2010.106>

## Fotocopiadoras

- Yuki Tsuchitoui and Hideki Sugiura. 2011. 10 MLOC in Your Office Copier. IEEE Softw. 28, 6 (November 2011), 93-95. DOI <http://dx.doi.org/10.1109/MS.2011.133>

## Automóviles

- Jürgen Mössinger: Software in Automotive Systems. IEEE Software 27(2):92-94, March/April 2010. DOI <http://dx.doi.org/10.1109/MS.2010.55>
- Les Hatton, Michiel van Genuchten: When Software Crosses a Line. IEEE Software 33(1):29-31, January/February 2016. DOI <http://dx.doi.org/10.1109/MS.2016.6>
- Matthias Traub, Alexander Maier, and Kai L. Barbehon. 2017. Future Automotive Architecture and the Impact of IT Trends. IEEE Software 34(3):27-32, May/June 2017. DOI <https://doi.org/10.1109/MS.2017.69>

## Aviones: Sistemas de gestión de vuelos

- David Avery. 2011. The Evolution of Flight Management Systems. IEEE Software 28, 1 (January 2011), 11-13. DOI <http://dx.doi.org/10.1109/MS.2011.17>

---

## Aviones: Software de control en la cabina del avión

- Stefan Burger, Oliver Hummel, and Matthias Heinisch. 2013. Airbus Cabin Software. IEEE Software 30, 1 (January 2013), 21-25. DOI <http://dx.doi.org/10.1109/MS.2013.2>

## Exploración petrolífera

- Michiel van Malckenhorst and Lex Mollinger. 2012. Going Underground. IEEE Software 29, 3 (May 2012), 17-20. DOI <http://dx.doi.org/10.1109/MS.2012.62>

## Exploración espacial

- Gerard J. Holzmann: Landing a Spacecraft on Mars. IEEE Software 30(2):83-86, March/April 2013. DOI <http://dx.doi.org/10.1109/MS.2013.32>
- Attila Baksa, Andras Balazs, Zoltan Pálos, Péter Spányi, Sandor Szalai, Laszlo Várhalmi: Software on a Comet: The Philae Lander's Central Onboard Computer. IEEE Software 33(2):81-84, March/April 2016. DOI <http://dx.doi.org/10.1109/MS.2016.40>

## Software científico: El experimento ATLAS

- David Rousseau. 2012. The Software behind the Higgs Boson Discovery. IEEE Software 29, 5 (September 2012), 11-15. DOI <http://dx.doi.org/10.1109/MS.2012.123>

## Software científico: Datos climáticos

- Nicholas Barnes and David Jones. 2011. Clear Climate Code: Rewriting Legacy Science Software for Clarity. IEEE Software 28, 6 (November 2011), 36-42. DOI <http://dx.doi.org/10.1109/MS.2011.113>

## Herramientas de desarrollo: Eclipse

- Zhihui Yang and Michael Jiang. 2007. Using Eclipse as a Tool-Integration Platform for Software Development. IEEE Software 24, 2 (March 2007), 87-89. DOI <http://dx.doi.org/10.1109/MS.2007.58>
- João Brunet, Gail C. Murphy, Dalton Serey, Jorge C. A. de Figueiredo: Five Years of Software Architecture Checking: A Case Study of Eclipse. IEEE Software 32(5):30-36, September/October 2015. DOI <http://dx.doi.org/10.1109/MS.2014.106>

- 
- Kim Moir. 2012. Eclipse. The Architecture of Open Source Applications. Volume II: Structure, Scale, and a Few More Fearless Hacks. <https://www.aosabook.org/en/eclipse.html>

### Herramientas de desarrollo: lint

- James Gimpel. 2014. Software That Checks Software: The Impact of PC-lint. IEEE Software 31, 1 (January 2014), 15-19. DOI <http://dx.doi.org/10.1109/MS.2014.13>

### Herramientas de desarrollo: Git

- Susan Potter. 2012. Git. The Architecture of Open Source Applications. Volume II: Structure, Scale, and a Few More Fearless Hacks. <https://www.aosabook.org/en/git.html>

### Herramientas de desarrollo: Automatización de pruebas

- Vahid Garousi and Frank Elberzhager. 2017. Test Automation: Not Just for Test Execution. IEEE Software 34(2):90-96, March/April 2017. DOI: <https://doi.org/10.1109/MS.2017.34>
- Ke Mao, Mark Harman, and Yue Jia. 2017. Robotic Testing of Mobile Apps for Truly Black-Box Automation. IEEE Software 34(2):11-16, March/April 2017. DOI: <https://doi.org/10.1109/MS.2017.49>
- Hiroaki Yoshida, Guodong Li, Takuki Kamiya, Indradeep Ghosh, Sreeranga P. Rajan, Susumu Tokumoto, Kazuki Munakata, Tadahiro Uehara. 2017. KLOVER: Automatic Test Generation for C and C Programs, Using Symbolic Execution. IEEE Software 34(5):30-37, September/October 2017. DOI <https://doi.org/10.1109/MS.2017.3571576>

### Herramientas de desarrollo: Desarrollo dirigido por modelos

- Patricia Lago, Ivano Malavolta, Henry Muccini, Patrizio Pelliccione, Antony Tang: The Road Ahead for Architectural Languages. IEEE Software 32(1):98-105, January/February 2015 DOI <http://dx.doi.org/10.1109/MS.2014.28>

### Herramientas de desarrollo: Verificación con Dafny

- K. Rustan M. Leino: Accessible Software Verification with Dafny. IEEE Software 34(6):94-97, November/December 2017. DOI <https://doi.org/10.1109/MS.2017.4121212>



---

## Herramientas de desarrollo: Selenium WebDriver

- Simon Stewart. 2012. Selenium WebDriver. The Architecture of Open Source Applications. Volume I: Elegance, Evolution, and a Few Fearless Hacks. <https://www.aosabook.org/en/selenium.html>

## Herramientas de desarrollo: LLVM

- Chris Lattner. 2012. LLVM. The Architecture of Open Source Applications. Volume I: Elegance, Evolution, and a Few Fearless Hacks. <https://www.aosabook.org/en/llvm.html>

## Herramientas de desarrollo: Paquetes en Python

- Tarek Ziadé. 2012. Python Packaging. The Architecture of Open Source Applications. Volume I: Elegance, Evolution, and a Few Fearless Hacks. <https://www.aosabook.org/en/packaging.html>

## Moodle

- Tim Hunt. 2012. Moodle. The Architecture of Open Source Applications. Volume II: Structure, Scale, and a Few More Fearless Hacks. <https://www.aosabook.org/en/moodle.html>

## nginx

- Andrew Alexeev. 2012. nginx. The Architecture of Open Source Applications. Volume II: Structure, Scale, and a Few More Fearless Hacks. <https://www.aosabook.org/en/nginx.html>

## Sendmail

- Eric Allman. 2012. Sendmail. The Architecture of Open Source Applications. Volume I: Elegance, Evolution, and a Few Fearless Hacks. <https://www.aosabook.org/en/sendmail.html>

## Seguridad: SQLMap (ataques por inyección de código SQL)

- Miroslav Stampar: "Heuristic methods used in sqlmap", FSec - FOI Security Symposium, Varazdin (Croatia), September 19th, 2013. <https://github.com/sqlmapproject/sqlmap/wiki/Presentations>
- Miroslav Stampar: "sqlmap - Under the Hood", PHDays in Moscow (Russia), May 23rd, 2013. <https://github.com/sqlmapproject/sqlmap/wiki/Presentations>

- 
- Angelo Ciampa, Corrado Aaron Visaggio, and Massimiliano Di Penta. 2010. A heuristic-based approach for detecting SQL-injection vulnerabilities in web applications. In Proceedings of the 2010 ICSE Workshop on Software Engineering for Secure Systems (SESS '10). ACM, New York, NY, USA, 43-49. DOI <http://dx.doi.org/10.1145/1809100.1809107>

## **Blockchain: Bitcoin**

- Arvind Narayanan and Jeremy Clark. 2017. Bitcoin's academic pedigree. Communications of the ACM 60(12):36-45, November 2017. DOI <https://doi.org/10.1145/3132259>

## **Blockchain: Ethereum (smart contracts)**

- Gavin Wood. 2016. Ethereum: A Secure Decentralised Generalised Transaction Ledger. <http://gavwood.com/paper.pdf>
- Qinghua Lu, Xiwei Xu. 2017. Adaptable Blockchain-Based Systems: A Case Study for Product Traceability. IEE Software 34(6):21-27, November/December 2017. DOI <https://doi.org/10.1109/MS.2017.4121227>

## **El núcleo de un sistema operativo**

- Clive King and Chris Beal. 2012. CSI Kernel: Finding a Needle in a Multiterabyte Haystack. IEEE Software 29, 6 (November 2012), 9-12. DOI <http://dx.doi.org/10.1109/MS.2012.154>

## **Cloud Computing: Contenedores (p.ej. Docker)**

- David Bernstein: Containers and Cloud: From LXC to Docker to Kubernetes. IEEE Cloud Computing 1(3):81-84, September 2014. DOI <http://dx.doi.org/10.1109/MCC.2014.51>
- Jürgen Cito, Gerald Schermann, John Erik Wittern, Philipp Leitner, Sali Zumberi, and Harald C. Gall. 2017. An empirical analysis of the docker container ecosystem on GitHub. In Proceedings of the 14th International Conference on Mining Software Repositories (MSR '17). IEEE Press, Piscataway, NJ, USA, 323-333. DOI <https://doi.org/10.1109/MSR.2017.67>

## **Internet de las cosas (IoT)**

- Michael Weyrich, Christof Ebert: Reference Architectures for the Internet of Things. IEEE Software 33(1):112-116, January/February 2016. DOI <http://dx.doi.org/10.1109/MS.2016.20>

- 
- Antero Taivalsaari, Tommi Mikkonen: A Roadmap to the Programmable World: Software Challenges in the IoT Era. IEEE Software 34(1):72-80, January/February 2017. DOI <http://dx.doi.org/10.1109/MS.2017.26>
  - Patrick Eugster, Vinaitheerthan Sundaram, Xiangyu Zhang: Debugging the Internet of Things: The Case of Wireless Sensor Networks. IEEE Software 32(1):38-49, January/February 2015. DOI <http://dx.doi.org/10.1109/MS.2014.132>

## **Automatización industrial**

- Stephan Pech. 2013. Software Agents in Industrial Automation Systems. IEEE Software 30, 3 (May 2013), 20-24. DOI [10.1109/MS.2013.57](http://dx.doi.org/10.1109/MS.2013.57)<http://dx.doi.org/10.1109/MS.2013.57>

## **Control de tráfico ferroviario**

- Koji Tomita and Keiichi Ito. 2011. Software in an Evolving Train Traffic Control System. IEEE Software 28, 2 (March 2011), 19-21. DOI <http://dx.doi.org/10.1109/MS.2011.45>

## **Sistemas de transporte automatizados**

- Danny Weyns and Michael Georgeff. 2010. Self-Adaptation Using Multiagent Systems. IEEE Software 27, 1 (January 2010), 86-91. DOI <http://dx.doi.org/10.1109/MS.2010.18>

## **Comercio electrónico: BestBuy.com**

- Joel Crabb: The BestBuy.com Cloud Architecture. IEEE Software 31(2):91-96 (2014). DOI <http://dx.doi.org/10.1109/MS.2014.45>

## **Servicios y microservicios en banca**

- Stephan Murer, Claus Hagen: Fifteen Years of Service-Oriented Architecture at Credit Suisse. IEEE Software 31(6):9-15 (2014). DOI <http://dx.doi.org/10.1109/MS.2013.137>
- Antonio Bucchiarone, Nicola Dragoni, Schahram Dustdar, Stephan T. Larsen, Manuel Mazzara: From Monolithic to Microservices: An Experience Report from the Banking Domain. IEEE Software, vol. 35, no. 3, pp. 50-55, May/June 2018. DOI <http://dx.doi.org/10.1109/MS.2018.2141026>

## Netflix

- Ali Basiri, Niosha Behnam, Ruud de Rooij, Lorin Hochstein, Luke Kosewski, Justin Reynolds, Casey Rosenthal: Chaos Engineering. IEEE Software 33(3):35-41, May/-June 2016. DOI <http://dx.doi.org/10.1109/MS.2016.60>

## eBay

- Rabih Bashroush and Eoin Woods. 2017. Architectural Principles for Energy-Aware Internet-Scale Applications. IEEE Software 34, 3 (May 2017), 14-17. DOI <https://doi.org/10.1109/MS.2017.60>